2001-1-27　J 研究会資料

J における有理数（分数）と浮動小数点数

連分数の変換と計算

西川　利男

Toshio.Nishikawa@kiu.ne.jp

このところ J によりいろいろな数論（整数を含めた）の問題を扱い，いまや楽しんでいる．数論と通常の科学技術計算との大きな違いは，有理数（rational number = 有比数），つまり分数としての計算にある．すなわち，分数では分母，分子をあらわに保持することにより，計算の途中で丸めを行うことなく，最終結果まで値の正確さを保っていくことができる．一方，浮動小数点計算では，たとえ多倍長精度によっても割り算による丸め誤差は避けられない．J では幸い r 形式による分数（有理数）計算を用いてこのようなことが可能である．

はじめに J でのデータ型についてまとめておこう．システム関数　3!:0 を作用させることで，返される結果により変数のデータ型が確認される．

1	boolean（論理の 0 か 1）	2	literal（文字，文字列）
4	integer （通常の整数）	8	floating（浮動小数点数）
16	complex （複素数）	32	boxed（ボックス型）
64	extended integer （拡張整数）	128	rational（有理数すなわち分数）

例えば，つぎのようになる．

```
   x0 =. 2
   3!:0 x0
4
   x1 =. 2x
   3!:0 x1
64
```

```
   x2 =. o.1
   x2
3.14159
   3!:0 x2
8
   x3 =. x: o.1
   x3
1285290289249r409120605684
   3!:0 x3
128
   x4 =. !100
   x4
9.33262e157
   3!:0 x4
8
   x5 =. !100x
   x5
933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761565182862536979208272237582511852109168640000000000000000000000000
   3!:0 x5
64
```

連分数（continued fraction）は分数の拡張版ともいえるものだが，Ｊによりきわめて効果的にこれに関する変換，計算が行える．なお，いわゆるペル方程式（２次元ディオファンタス不定方程式）の解が連分数によりえられるが，これについてはあらためて報告する予定である．

連分数はつぎのようにして作られる．例えば

$$\frac{19}{7} = 2 + \frac{1}{7/5} = 2 + \frac{1}{1 + \frac{1}{5/2}} = 2 + \frac{1}{1 + \frac{1}{2 + \frac{1}{2}}}$$

これは表記の便のためつぎのように表す.

(2 1 2 2)

一方，上のような連分数表記をもとにその値を得ることも必要である.

Jにおいては，ごく簡単に

```
   (+%)/2 1 2 2
2.71429
```

として得られる．しかし，これをr形式として得るにはちょっとした工夫が必要である.

以上のような連分数の変換などを行ういくつかのJの関数を作った.

```
NB. 通常分数を連分数に変換する
NB. 仮分数を帯分数にする  mixfr 19 7 -> 2 7 5
mixfr =: 3 : '((({.y.)-r)%d),(d=. {:y.),r=. |~/y.'
NB. makecont 19 7 -> 2 1 2 2
makecont =: 3 : 0
y =. mixfr y.
while. 0~: {: y
  do. y =. (_2}.y), mixfr _2{.y
  end.
_2}.y
)

NB. 連分数の値を得る
NB.  monadic -- 分数表示
NB.   e,g. contx 2 2 1 5 -> 40r17
NB.  dyadic  -- 浮動小数表示
NB.   e.g. 1 contx 2 2 1 5 -> 2.35294118 (default)
NB.   e.g. 16j14 contx 2 2 1 5 -> 2.35294117647059
contx =: 3 : 0
r =. ". '1r',":{:y.
```

```
p =. }: y.
q =. {: p
q =. q + r
p =. (}:p), q
s =. (+%)/p
:
if. 0 = {: +. x. do. x. =. 10j8 end.
x. ": contx y.
)
```

以下，実行例をあげる．
関数 makecont は通常分数を連分数へ変換する．

```
   makecont 19 7
2 1 2 2
```

連分数に対して，関数 contx を１価関数として使ったときは r 形式の通常分数に変換する．

```
   contx 2 1 2 2
19r7
```

関数 contx を２価関数として使うと浮動小数点数の値が返される．

```
   1 contx 2 1 2 2
2.71428571
```

なお，左引数を j 形式により精度を指定できる． j 形式でないときは（任意の整数）デフォルトの精度がとられる．

連分数表現は無理数に対しても有効であり，この場合は無限に続く連分数になる．
例として$\sqrt{2}$はつぎのような無限項連分数で表わせる．

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cdots}}}}$$

これを少しずつ項を増やしてその値を見てみよう.

```
        contx 1 2 2
7r5
        contx 1 2 2 2
17r12
        contx 1 2 2 2 2
41r29
        20j18 contx 1 2 2 2 2 2
1.414285714285714286
        20j18 contx 1 2 2 2 2 2 2
1.414201183431952663
        20j18 contx 1 2 2 2 2 2 2 2
1.414215686274509804
        20j18 contx 1 2 2 2 2 2 2 2 2
1.414213197969543147
        20j18 contx 1 2 2 2 2 2 2 2 2 2
1.414213624894869638
        20j18 contx 1 2 2 2 2 2 2 2 2 2 2
1.414213551646054694
        20j18 contx 1 2 2 2 2 2 2 2 2 2 2 2
1.414213564213564214
        20j18 contx 1 2 2 2 2 2 2 2 2 2 2 2 2
1.414213562057320463
```